Cincom

AD/ADVANTAGE

MANTIS 2.8.01 Installation and Startup
OpenVMS/UNIX

P39-0027-00

AD/Advantage® MANTIS 2.8.01 Installation and Startup OpenVMS/UNIX

Publication Number P39-0027-00

Cincom Systems, Inc.
55 Merchant Street
Cincinnati, Ohio 45246-3732
U. S. A.

PHONE: (513) 612-2300
FAX: (513) 612-2000
WORLD WIDE WEB: http://www.cincom.com

Attention:

Some Cincom products, programs, or services referred to in this publication may not be available in all countries in which Cincom does business. Additionally, some Cincom products, programs, or services may not be available for all operating systems or all product releases. Contact your Cincom representative to be certain the items are available to you.

Release information for this manual

The *AD/Advantage MANTIS 2.8.01 Installation and Startup OpenVMS/UNIX*, P39-0027-00, is dated February 12, 2001. This document supports Release 2.8 of MANTIS.

We welcome your comments

We encourage critiques concerning the technical content and organization of this manual. Please take the survey provided with the online documentation at your convenience.

Cincom Technical Support for AD/Advantage

FAX: (513) 612-2000
Attn: MANTIS Support

E-mail: helpna@cincom.com

Phone: 1-800-727-3525

Mail: Cincom Systems, Inc.
Attn: MANTIS Support
55 Merchant Street
Cincinnati, OH 45246-3732
U. S. A.

Contents

About this book

Using this document

MANTIS® is an application development system that consists of design facilities (e.g., screens and files) and a programming language. This manual is intended for the AD/Advantage system administrator, who is expected to be familiar with use and administration of the UNIX® operating system.

Document organization

The information in this manual is organized as follows:

Chapter 1—Installing AD/Advantage—MANTIS
Contains procedures and considerations for installing AD/Advantage.

Chapter 2—MANTIS run-time environment commands
Describes the MANTIS run-time environment commands.

Appendix A—Sample MANTIS installation (minstall) output
Shows the output from an actual MANTIS installation under SCO UNIX.

Appendix B—MANTIS for UNIX authorization
Provides a brief introduction to the MANTIS security system, which controls the use of MANTIS and its components through an encrypted security code.

Appendix C—Platform-specific installation notes
Contains installation considerations for SCO UNIX and SUN-OS environments.

Appendix D—New features and changes in service levels
Describes the new features and changes in service level 2801.

Index

Revisions to this manual

The following changes have been made for this release.

- Updated Publication Release Number from P19-0027-03 to P39-0027-00.
- Updated publication titles and numbers under “MANTIS documentation series" on page xiv and the entire document.
- Updated MANTIS Patch information in “Giving users access to a MANTIS class” on page 29.
- Updated MANTIS Patch screen information in “Writing a class SETUP script” starting on page 31.
- Updated Test Sign on and Facility Selection screens in “Testing the MANTIS class” starting on page 36.
- Updated MANTIS patch information under “MANTIS for UNIX authorization” on page 55.
- Deleted reference to Service Level 2406-2410 and 2592 in “New features and changes in service levels” starting on page 59.
- Added information in “Service level 2801” on page 71.

Conventions

The following table describes the conventions used in this document series:

Convention	Description	Example
Constant width type	Represents screen images and segments of code.	Screen Design Facility GET NAME LAST INSERT ADDRESS
Slashed b (⍽)	Indicates a space (blank). The example indicates that a password can have a trailing blank.	WRITEPASS⍽
Brackets [ ]	Indicate optional selection of parameters. (Do not attempt to enter brackets or to stack parameters.) Brackets indicate one of the following situations.	
	A single item enclosed by brackets indicates that the item is optional and can be omitted. The example indicates that you can optionally enter a program name.	COMPOSE [*program-name*]
	Stacked items enclosed by brackets represent optional alternatives, one of which can be selected. The example indicates that you can optionally enter NEXT, PRIOR, FIRST, or LAST. (NEXT is underlined to indicate that it is the default.)	[**NEXT** **PRIOR** **FIRST** **LAST**]
Braces { }	Indicate selection of parameters. (Do not attempt to enter braces or to stack parameters.) Braces surrounding stacked items represent alternatives, one of which you must select. The example indicates that you must enter FIRST, LAST, or a value for *begin*.	{**FIRST** ***begin*** **LAST**}

Convention	Description	Example
Underlining (In syntax)	Indicates the default value supplied when you omit a parameter. The example indicates that if you do not specify ON, OFF, or a row and column destination, the system defaults to ON.	**SCROLL** [**ON** | **OFF** | [**row**][**, col**]]
	Underlining also indicates an allowable abbreviation or the shortest truncation allowed. The example indicates that you can enter either PRO or PROTECTED.	PROTECTED
Ellipsis points...	Indicate that the preceding item can be repeated. The example indicates that you can enter (A), (A,B), (A,B,C), or some other argument in the same pattern.	(*argument*,...)
UPPERCASE	Indicates MANTIS reserved words. You must enter them exactly as they appear. The example indicates that you must enter CONVERSE exactly as it appears.	CONVERSE *name*
Italics	Indicate variables you replace with a value, a column name, a file name, and so on. The example indicates that you can supply a name for the program.	COMPOSE [*program-name*]
Punctuation marks	Indicate required syntax that you must code exactly as presented. ( ) parentheses . period , comma : colon ' ' single quotation marks	[**LET**] *v* [(*i*) | (*i*, *j*)] [**ROUNDED**(*n*)] = *e1* [, *e2*, *e3*...]

Convention	Description	Example
UNIX OpenVMS	Information specific to a certain operating system is flagged by a symbol in a shadowed box (e.g., UNIX) indicating which operating system is being discussed. Skip any information that does not pertain to your environment.	UNIX DBA will run on any terminal that supports the cursor library.

MANTIS documentation series

MANTIS is an application development system designed to increase productivity in all areas of application development, from initial design through production and maintenance. MANTIS is part of AD/Advantage®, which offers additional tools for application development. Below are listed the manuals offered with MANTIS in the OpenVMS™ and UNIX environments, organized by task. You may not have all the manuals that are listed here. For a synopsis of each manual, refer to the *AD/Advantage MANTIS Application Development Tutorial OpenVMS/UNIX*, P39-1340.

Getting started

- *AD/Advantage MANTIS 2.8.01 Installation and Startup OpenVMS/UNIX*, P39-0027*

General use

- *AD/Advantage MANTIS Facilities OpenVMS/UNIX*, P39-1300*
- *AD/Advantage MANTIS Language OpenVMS/UNIX*, P39-1310
- *AD/Advantage MANTIS Messages and Codes OpenVMS/UNIX*, P39-1330*
- *AD/Advantage MANTIS Application Development Tutorial OpenVMS/UNIX*, P39-1340
- *AD/Advantage MANTIS SUPRA SQL Programming OpenVMS/UNIX*, P39-1345
- *AD/Advantage MANTIS Rdb Programming UNIX*, P39-1350
- *AD/Advantage MANTIS Oracle Programming UNIX*, P39-1355

Manuals marked with an asterisk (*) are listed twice because you use them for different tasks.

Master User tasks

- *AD/Advantage MANTIS Facilities OpenVMS/UNIX*, P39-1300*
- *AD/Advantage MANTIS Administration OpenVMS/UNIX*, P39-1320
- *AD/Advantage MANTIS 2.8.01 Installation and Startup OpenVMS/UNIX*, P39-0027*

Manuals marked with an asterisk (*) are listed twice because you use them for different tasks.

Educational material

MANTIS educational material is available from your regional Cincom education department.

1

Installing AD/Advantage—MANTIS

The very first step in the installation process, is to scan the table of contents of this document, looking for any relevant entries listed under “Platform-specific installation notes” on page 57. If relevant entries are found, they should be read carefully prior to beginning the installation process.

Installing AD/Advantage and a new MANTIS system

Disk space requirements for AD/Advantage

The physical AD/Advantage and MANTIS will require approximately 20 MB of disk space. Each additional AD/Advantage class will require a minimum of 3.5 MB of additional disk space.

Downloading the AD/Advantage media

The physical AD/Advantage distribution media will indicate what format the tape is in. For UNIX this will either be tar or cpio format. Use the label on your distribution media to determine which of the following sections is applicable to your release media.

Downloading AD/Advantage from tar format media

The following commands can be used to download the tar format AD/Advantage media, into the existing MANTIS system ($MANTIS_ROOT is assumed to be defined properly):

```
$ cd $MANTIS_ROOT
$ –tar ~ xovf < device-name
```

You must supply an appropriate *device-name*.

This *tar* command will automatically add the AD/Advantage files into the current MANTIS disk structure. All files will be located in the following subdirectories of *mantis_root*:

Directory name	Primary contents
$MANTIS_ROOT/bin	Shell scripts
$MANTIS_ROOT/lib	Support files
$MANTIS_ROOT/exe	Executable files
$MANTIS_ROOT/data	Data files

REMINDER: MANTIS_ROOT is a UNIX environment variable that points to the mantis_root directory. Using the example where mantis_root is /cincom/mantis_2801, $MANTIS_ROOT/bin actually refers to a physical directory name of /cincom/mantis_2801/bin.

For floppies, tar will occasionally stop reading in the middle of a set of floppies. This happens when a file ends very near the end of a floppy, and tar mistakenly thinks it is the last volume in the media set. If this should happen, you can reissue the tar command and start with the next volume in the installation media set. For example, if tar stops after volume 2 of the media set, you could reissue the tar command with volume 3 in the drive.

Downloading AD/Advantage from cpio format media

The following commands can be used to download the cpio format AD/Advantage media, into the existing MANTIS system ($MANTIS_ROOT is assumed to be defined properly):

```
$ cd $MANTIS_ROOT
$ cpio -icvd < device-name
```

You must supply an appropriate *device-name*.

This cpio command will automatically add the AD/Advantage files into the current MANTIS disk structure. All files will be located in the following subdirectories of $MANTIS_ROOT:

Directory name	Primary contents
$MANTIS_ROOT/bin	Shell scripts
$MANTIS_ROOT/lib	Support files
$MANTIS_ROOT/exe	Executable files
$MANTIS_ROOT/data	Data files

REMINDER: MANTIS_ROOT is a UNIX environment variable that points to the mantis_root directory. Using the example where mantis_root is /cincom/mantis_2801, $MANTIS_ROOT/bin actually refers to a physical directory name of /cincom/mantis_2801/bin.

Installing the AD/Advantage—MANTIS files

Once the files have been downloaded to disk, the next step is to install the AD/Advantage—MANTIS system. To accomplish this, perform the following steps:

1. Change working directory to the $MANTIS_ROOT directory:

   ```
   # cd $MANTIS_ROOT
   ```

2. Invoke the AD/Advantage installation procedure:

   ```
   # sh ./minstall 2>&1 | tee minstall.log
   ```

   (This command will run the minstall script from the current directory, and send the output both to the screen, and to a file called minstall.log. The minstall.log file can be used to debug any problems that should arise.)

3. The adv_inst script will automatically install MANTIS and the layered AD/Advantage components. A number of messages will be printed out during this process. A log of the complete installation process will be kept on the minstall.log file. For examples of output from the minstall script for a new AD/Advantage—MANTIS system, see “Sample MANTIS installation (minstall) output” on page 53.

   One of the steps that MANTIS installation (minstall) script performs is to compile a terminfo database that describes the capabilities of various terminals. Some versions of UNIX do not support all terminal capabilities. It is possible that you will see several messages similar to the following:

   ```
   tic:  WARNING:  near line 95:  terminal...
   ```

   These errors occur on some platforms, due to limitations in the TERMINFO facilities. These tic warning messages indicate that the operating system you are running is incapable of supporting some terminal features. These messages can be safely ignored in most cases.

Upgrading an older version of a MANTIS cluster

When upgrading from an older version of a MANTIS cluster, you may use the “mcpy” (MANTIS copy) utility to upgrade a MANTIS Release 2701 Cluster format to MANTIS Release 2801 Cluster format. If you are upgrading from a release of MANTIS that is older than MANTIS Release 2701, then you must use the UEF (Universal Export Facility) to export all MANTIS entities from your older MANTIS cluster into a new MANTIS Release 2801 cluster. To use the MANTIS Copy (mcpy) utility, perform the following steps:

1. Have your $MANTIS_ROOT environment variable set to the new MANTIS Release Directory.

2. Set the PATH environment variable to include the path to $MANTIS_ROOT/bin directory, for example:

   ```
   export PATH=$MANTIS_ROOT/bin:$PATH
   ```

3. Set the MANTIS_CLASS environment variable to 2801, for example:

   ```
   export MANTISCLASS=2801
   ```

4. Change directories (cd) to the MANTIS Data directory, where all MANTIS Clusters are contained for this release of MANTIS, for example:

   ```
   cd $MANTIS_ROOT/data
   ```

5. At the Unix prompt enter mcpy. This utility will ask for the following items:

   - Input Cluster Name,
   - Release Cluster Name,
   - Output Cluster Name
   - Type of MANTIS Copy

The following example of the mcpy utility shows the upgrade of MANTIS Release of 2701 with a 2701 Cluster name of “man_PR27” to MANTIS Release 2801, with a new MANTIS Cluster Name of “man_PR28”.

```
Enter Input MANTIS/Transfer File:/cincom/mantis_2701/data/man_PR27
Enter Release MANTIS/Transfer File :  man_2801
Enter Output MANTIS/Transfer File :  man_PR28
Upgrade (U), Sidegrade (S), Downgrade (D), please specify.
Enter:  U, S, D, U
```

6. Once the mcpy utility has finished, you must create a new MANTIS Class File for the new Cluster Name. The following example uses the new Cluster Name of man_PR28.

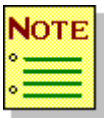
   At the UNIX prompt, enter the following:

   ```
   cp_class 2801 PR28 2801
   ```

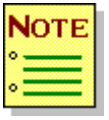
   The above cp_class utility will then build a man_PR28 file out in the $MANTIS_ROOT/bin directory. At this point you can now reset your MANTIS_CLASS environment variable to point to the new MANTIS Cluster of man_PR28, for example:

   ```
   export MANTIS_CLASS=PR28
   ```

   Now when you bring MANTIS Release 2801 up, you will be pointing to the new MANTIS Cluster (man_PR28) containing all of you MANTIS entities from your old MANTIS Cluster (man_PR27) (Users, Programs, Screens, Views, etc.)



The “mcpy” utility WILL NOT copy any entities from the “MASTER” User into the new MANTIS Cluster. So if you added any entities to the “MASTER” User (such as Programs, Screens, Views, etc.) then these entities will have to be exported out of the old MANTIS Cluster and into the new MANTIS Cluster for the “MASTER” User using the Universal Export Facility (UEF).



The MANTIS Clusters and Transfer Files contained within the $MANTIS_ROOT/data directory (man_2801, man_ADA2, xfr_2801 and xfr_ADA2) should not be used as production Clusters and Transfer Files. These files are to be used as a “Template” to create new Clusters and Transfer Files. This will then allow you to create new MANTIS Clusters and Transfer Files from various MANTIS utility programs, incase your files and nightly backup get corrupted.

Introduction to the MANTIS working environment

After you have installed the MANTIS system, it is time to start ongoing administration tasks. The first task is to create a MANTIS class or set of classes for the end users. The following introductory material is essential to understanding this process.

MANTIS classes

MANTIS users are divided into groups called MANTIS classes. All members of a given class share two groups of physical files:

- **MANTIS File.** Large storage area that holds all MANTIS entities (programs, screens, file definitions, etc.) available to the users of that class. The MANTIS File is implemented under UNIX as a MISAM file (see the next section, "MISAM files").

- **MANTIS Transfer File.** Temporary storage area used to move data between users or classes. A Transfer File can be shared by multiple classes. The Transfer File is implemented under UNIX as a MISAM file (see the next section, "MISAM files").

MISAM files

The UNIX operating system does not include a general-purpose, indexed, sequential file system. Because MANTIS is built internally around indexed files, it provides such a file system as part of the MANTIS package. The MANTIS indexed file system is called MISAM (MANTIS Indexed Sequential Access Method).

A MISAM file is actually a group of related files that the MISAM file system uses to represent a single, logical, indexed sequential file. A MISAM file can have any name, but each of the component files has one of the following suffixes added to the filename: *.nx*, *.st*, *.ki*, or *.mp*. Each file contains different kinds of data or metadata, and all are necessary to the MISAM system.

A UNIX *ls* command issued against the $MANTIS_ROOT/data directory shows the following series of files:

- man_2801.st
- man_2801.nx
- man_2801.ki
- man_2801.mp.

These four UNIX files are a single “empty” MANTIS file.

MISAM file-naming conventions

$MANTIS uses a file-naming convention for all MANTIS files as follows:

```
man_cccc.ss
```

where:

cccc is the 1–4 character name of the MANTIS Cluster File or the MANTIS Transfer File. The MANTIS class environment variable is used to set the name of the MANTIS Cluster File. The MANTIS transfer environment variable is used to set the name of the MANTIS Transfer File.

ss is the 2-character file suffix described in “MISAM files” on page 23.

For example, the four files that compose the MANTIS File for the TEST class would be named as follows:

- man_TEST.st
- man_TEST.mp
- man_TEST.nx
- man_TEST.ki

All MANTIS MISAM files are stored in the $MANTIS_ROOT/data directory.

MANTIS run-time environment administration commands organization

MANTIS run-time environment administration commands are implemented as shell scripts and are stored in the $MANTIS_ROOT/bin directory. Each administration command is implemented as a separate script that can be invoked directly from the UNIX prompt. After becoming familiar with the various commands, the MANTIS administrator may want to call these shell scripts directly. However, MANTIS includes a menu shell script called mant_mgr that interactively assists the administrator in building the calling arguments for the administration command shell scripts. We recommend that a new administrator use mant_mgr.

MANTIS run-time environment administration commands are described in “MANTIS run-time environment commands” on page 43.

Creating your first MANTIS class

MANTIS is delivered with a clean MANTIS class (called 2801) that is copied to make working classes. MANTIS classes should never use the clean MANTIS File and the clean MANTIS Transfer File (called xfr). These files will be needed as templates for all new classes that are created in the future. If these files are inadvertently updated, you will need to reinstall MANTIS from the distribution media.

Running either the cp_class command or the interactive mant_mgr script, which calls cp_class, creates a new class. The following screens show how to use mant_mgr to create the TEST class. (For information on using the mant_mgr script, see "MANTIS run-time environment commands" on page 43.)

1. Start the mant_mgr script:

```
# mant_mgr
```

The main menu displays:

```
Option  ( Mantis Command ) Description
------  ---------------------------------------------------

0 ... ( mant_mgr  ) Description of this MENU's functions
1 ... ( cp_class  ) Copy an existing MANTIS CLASS
2 ... ( rm_class  ) Delete a MANTIS class
3 ... ( cp_misam  ) Copy a MANTIS MISAM file
4 ... ( rm_misam  ) Delete a MANTIS MISAM file
5 ... ( mcpy      ) Upgrade MANTIS or Merge two MANTIS files
6 ... ( interface ) Select MANTIS internal interface version
7 ................. Display MANTIS directories
8 ... ( mant      ) Run MANTIS
9 ... ( mk_setup  ) Generate a USER setup file for a MANTIS class

10 ............... Exit

Please enter the option number and press RETURN:
```

2. Choose option 1, Copy an existing MANTIS class. The following screen displays:

```
cp_class:

The MANTIS Environment uses the concept of MANTIS CLASSES
to group related users together.  Members of a class share
a single MANTIS STORAGE area known as a MISAM file, in which
all MANTIS entities (screens, programs files...) are stored.

Every MANTIS user must belong to a MANTIS CLASS.  MANTIS
examines the MANTIS_CLASS environment variable to determine
which class a user belongs to.

MANTIS is delivered with a CLEAN CLASS which is named the same as
the MANTIS service level (e.g., 2801).  The clean class is copied to
produce other working classes.  For this reason, the clean class
should never be used to store user data.

The available old MANTIS class(es) are :

2801

Please Enter the source MANTIS class name :  [] 2801
Please Enter the new class name  :  [] TEST
Do you want the new class to have a different
MANTIS transfer file ? y|n  [y] y
Please Enter new MANTIS transfer file name :  [] TXFR
.......Please press return to go back to the main menu
```

a. At the first prompt, enter the source MANTIS class name, the name of the MANTIS class you want to copy.

b. At the second prompt, enter the new class name, the name of the MANTIS class you want to create.

c. If you want the new class to have a different Transfer File, enter Y at the third prompt. If you enter Y, you are then prompted for the name of the new Transfer File.

3. Press RETURN to go back to the main menu.

```
Option  ( Mantis Command ) Description
------  ---------------------------------------------------

0 ... ( mant_mgr  ) Description of this MENU's functions
1 ... ( cp_class  ) Copy an existing MANTIS CLASS
2 ... ( rm_class  ) Delete a MANTIS class
3 ... ( cp_misam  ) Copy a MANTIS MISAM file
4 ... ( rm_misam  ) Delete a MANTIS MISAM file
5 ... ( mcpy      ) Upgrade MANTIS or Merge two MANTIS files
6 ... ( interface ) Select MANTIS internal interface version
7 ................. Display MANTIS directories
8 ... ( mant      ) Run MANTIS
9 ... ( mk_setup  ) Generate a USER setup file for a MANTIS class

10 ............... Exit

Please enter the option number and press RETURN:
```

4. Choose option 10, Exit.

The new MANTIS class is now available for use.

Giving users access to a MANTIS class

Before starting MANTIS, the following environment variables must be exported to each user's UNIX environment. (For information on setting up your user environment, see "Writing a class SETUP script" on page 31.)

- **MANTIS_CLASS.** Must contain the name of the MANTIS class of which the current user is a member. An example class named TEST is used throughout this guide. A command to set up the TEST MANTIS_CLASS could look like:

```
# MANTIS_CLASS=TEST; export MANTIS_CLASS
```

- **MANTIS_ROOT.** Must point to the $MANTIS_ROOT directory—for example:

```
# export MANTIS_ROOT=/cincom/mantis_2801
```

- **MANTIS_PATCH.** Must contain the encrypted security (authorization) patch that was delivered with the product installation media. In order for MANTIS to run, you must replace the dummy information in this command with the security patch information delivered with your tapes.

```
# MANTIS_PATCH='11111111  22222222  33333333  44444444
  5555555  66666666  77777777  88888888'
# export MANTIS_PATCH
```

- **MANTIS_TERM.** *Optional*. Contains the name of the terminal that is being used to access MANTIS. If this variable does not exist, MANTIS uses the contents of the UNIX TERM variable. MANTIS_TERM is provided to avoid terminal naming conflicts between MANTIS and other products.

```
# export MANTIS_TERM=vt100
```

- **MANTIS_TERMINFO.** *Optional*. Overrides the MANTIS default location of the MANTIS TERMINFO database ($MANTIS_ROOT/data). It is only necessary to specify MANTIS_TERMINFO if the TERMINFO database supplied with MANTIS has been altered.

```
# MANTIS_TERMINFO=$MANTIS_ROOT/data
# export MANTIS_TERMINFO
```

To control automatic connections to SUPRA Server SQL databases, you can set up the following variables:

- **MANTIS_SQL_DBNAME.** *Optional*. Should be set to the name of the SUPRA Server SQL database that will be used.

  ```
  # export MANTIS_SQL_DBNAME=TESTDB
  ```

- **MANTIS_SQL_USER.** *Optional*. Should be set to the SUPRA Server SQL user that will be used.

  ```
  # export MANTIS_SQL_USER=TEST
  ```

- **MANTIS_SQL_PASS.** *Optional*. Should be set to the password for the MANTIS_SQL_USER.

  ```
  # export MANTIS_SQL_PASS=TESTPASS
  ```

  In addition, the $MANTIS_ROOT/bin directory must be included in the user's PATH. All MANTIS administration and execution commands are accessed via shell scripts that are located in this directory. On most systems, this can be accomplished by adding lines similar to the following to each user's profile:

  ```
  # PATH=$MANTIS_ROOT/bin:$PATH
  # export PATH
  ```

Writing a class SETUP script

You can use the mk_setup option of mant_mgr to generate a sample setup script for a class. mk_setup uses the current UNIX process environment when it is called. If this is the first class, mk_setup will generate a file that has commented lines for each of the possible MANTIS SETUP environment variables. The generated file is named

*cccc*_SETUP

where:

cccc is your 4-digit class name.

mant_mgr moves the setup script into the $MANTIS_ROOT/bin directory. You must edit the setup file using a standard text editor to fill in the correct values for each environment variable and to uncomment the necessary lines. To edit the setup file, perform the following steps:

1. Start the mant_mgr script:

```
# mant_mgr
```

The main menu displays:

```
Option  ( Mantis Command ) Description
------  ---------------------------------------------------

0 ... ( mant_mgr  ) Description of this MENU's functions
1 ... ( cp_class  ) Copy an existing MANTIS CLASS
2 ... ( rm_class  ) Delete a MANTIS class
3 ... ( cp_misam  ) Copy a MANTIS MISAM file
4 ... ( rm_misam  ) Delete a MANTIS MISAM file
5 ... ( mcpy      ) Upgrade MANTIS or Merge two MANTIS files
6 ... ( interface ) Select MANTIS internal interface version
7 ................. Display MANTIS directories
8 ... ( mant      ) Run MANTIS
9 ... ( mk_setup  ) Generate a USER setup file for a MANTIS class

10 ............... Exit

Please enter the option number and press RETURN:
```

2. Choose option 9, Generate a user setup file for a MANTIS class. The following screen displays:

```
The mk_setup procedure reads the current process environment
and generates a setup file that can be executed to give a
process access to a given class.

It reads the current settings for the ENVIRONMENT VARIABLES:

MANTIS_ROOT    MANTIS_CLASS   MANTIS_SQL_DBNAME
MANTIS_SECURITY    MANTIS_SQL_USER    MANTIS_SQL_PASS

and uses their contents to generate a setup script.

The following classes are currently available:

2801    TEST

Please enter the MANTIS class name :  [TEST]
```

3. Enter the name of the MANTIS class in the field. The following screen displays:

```
WARNING:  The MANTIS_PATCH environment variable is not currently
defined.  This is required for MANTIS to operate.
The startup script is being generated with a DUMMY definition
for MANTIS_PATCH, that must be edited and corrected with the
SECURITY codes that were included with your MANTIS product.

The generate script will not work, until these codes are corrected
by hand editing, or regenerating the script.

File TEST_SETUP created in directory:

/cincom/mantis_2801/bin

This file can be executed by use of the command:

. /cincom/mantis_2801/bin/TEST_SETUP


(NOTE the DOT (.) and space are a required part of this command)

The script can executed from the UNIX prompt to gain temporary
access to the TEST  MANTIS class, or from a .profile script to allow
permanent access to the TEST class.  TEST_SETUP must be executed by
each user before that user will be able to access the TEST class.

.......Please press return to go back to the main menu
```

The next step is to edit the output SETUP script and set up the necessary security codes, so if the warning messing shown in the preceding screen should appear, it can be safely ignored.

4. Press RETURN to go back to the main menu.

```
Option  ( Mantis Command ) Description
------  ---------------------------------------------------

0 ... ( mant_mgr  ) Description of this MENU's functions
1 ... ( cp_class  ) Copy an existing MANTIS CLASS
2 ... ( rm_class  ) Delete a MANTIS class
3 ... ( cp_misam  ) Copy a MANTIS MISAM file
4 ... ( rm_misam  ) Delete a MANTIS MISAM file
5 ... ( mcpy      ) Upgrade MANTIS or Merge two MANTIS files
6 ... ( interface ) Select MANTIS internal interface version
7 ................. Display MANTIS directories
8 ... ( mant      ) Run MANTIS
9 ... ( mk_setup  ) Generate a USER setup file for a MANTIS class

10 ............... Exit

Please enter the option number and press RETURN:
```

5. Choose option 10, Exit.

6. To edit the generated script, enter the following:

```
#     vi $MANTIS_ROOT/bin/TEST_SETUP
```

The following screen displays

```
MANTIS_CLASS=TEST
export MANTIS_CLASS

MANTIS_ROOT=/cincom/mantis_2801
export MANTIS_ROOT

# MANTIS_PATCH not defined in current environment.
# including a dummy definition, that must be edited
# before this class can be used.

#MANTIS_PATCH="xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
#xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx"
#export MANTIS_PATCH

#MANTIS_TERM=xxxxx
#export MANTIS_TERM

#MANTIS_TERMINFO=xxxxxx
#export MANTIS_TERMINFO

#MANTIS_SQL_DBNAME=xxxxx
#export MANTIS_SQL_DBNAME

#MANTIS_SQL_USER=xxxxx
#export MANTIS_SQL_USER

#MANTIS_SQL_PASS=xxxxx
#export MANTIS_SQL_PASS

#MANTIS_INIT=xxxxx
#export MANTIS_IMIT

PATH=$MANTIS_ROOT/bin:$PATH
```

7. Use the editor to enter the needed values and uncomment the needed lines. For the TEST class, the resultant file looks something like:

```
MANTIS_CLASS=TEST
export MANTIS_CLASS

MANTIS_ROOT=/cincom/mantis_2801
export MANTIS_ROOT

MANTIS_PATCH="11111111 22222222 33333333 44444444 55555555 66666666 77777777 88888888″
export MANTIS_PATCH

#MANTIS_TERM=xxxxx
#export MANTIS_TERM

#MANTIS_TERMINFO=xxxxxx
#export MANTIS_TERMINFO

MANTIS_SQL_DBNAME=TESTDB
export MANTIS_SQL_DBNAME

MANTIS_SQL_USER=TEST
export MANTIS_SQL_USER

MANTIS_SQL_PASS=TESTPASS
export MANTIS_SQL_PASS

#MANTIS_INIT=xxxxx
#export MANTIS_IMIT

PATH=$MANTIS_ROOT/bin:$PATH
```

8. All users who are going to use the TEST MANTIS class would add the following to their profile:

```
#  Now run the MANTIS TEST CLASS SETUP PROCEDURE
#  In the current shell  NOTE THE dot (". ") and the
#  space in this command:

. /cincom/mantis_2801/bin/TEST_SETUP
```

After you have a working MANTIS system, the process of generating new SETUP scripts will be much simpler. When you run mant_mgr, it will read the various environment variable settings in your current user and generate live lines, rather than commented lines. After your first class, editing the SETUP files should be minimal.

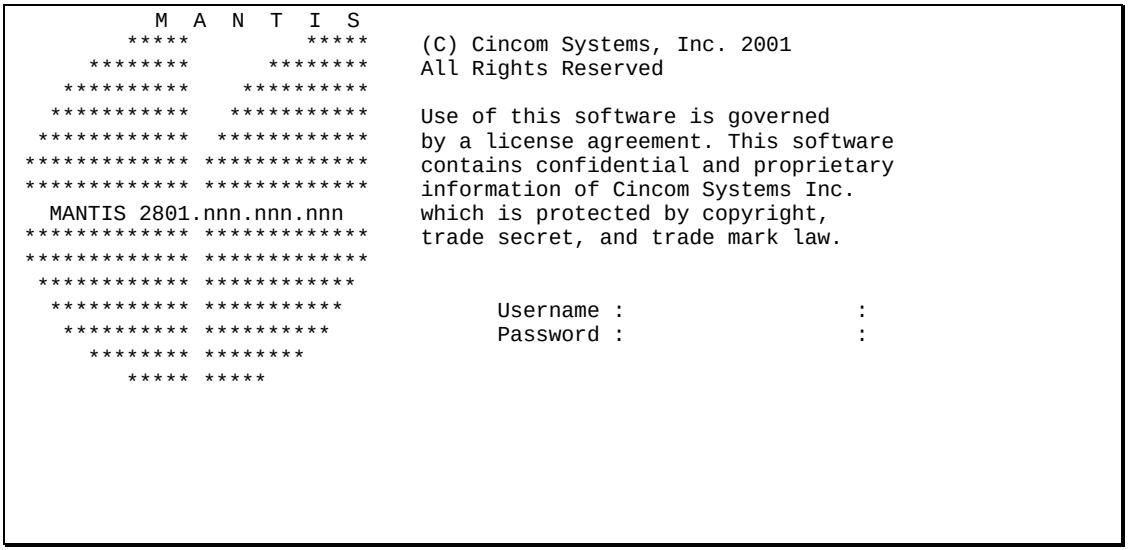
Testing the MANTIS class

You should test these new procedures by invoking them in the MANTIS administrator's account before giving them to users. The following steps show how this is accomplished for the TEST class:

1. Enter the following:

```
# .   /cincom/mantis_2801/bin/TEST_SETUP
# mant
```

The screen below displays:



2. Enter a user name and password. Refer to *AD/Advantage MANTIS Administration Open VMS/Unix*, P39-1320, which provides instructions on setting up user accounts. To maintain the MANTIS environment, the MANTIS administrator will use a MANTIS user named MASTER (initial password MASTER) to create and update accounts and to select run-time options for each class.

   A menu similar to the following displays:

```
                          M  A  N  T  I  S

                       FACILITY SELECTION

Run a Program  ..............  1        Transfer Facility ......... 12
Display a Prompter  .........  2        Edit MANTIS Messages....... 13
Design a Program  ...........  3        Directory Facility ........ 14
Design a Screen  ............  4        Universal Export Facility.. 15
Design a File Profile........  5        Update Shared Entity List.. 16
Design a Prompter  ..........  6        Update Language Codes  .... 17
Design a User Profile .......  7        MANTIS Maintenance  ....... 18
Design an Interface .........  8        Spectra  .................. 19
Design An Ultra File View ...  9        Search Facility   ......... 20
Design An External File View 10        List of Mantis Users ...... 21
Sign On As Another User......11        MANTIS Security Patch Info. 22
                                    Exit MANTIS ............... CANCEL



                          :    :
```

On most terminals, you can exit this menu by pressing CTRL-Z.

UNIX security and the MANTIS_FILE

The MANTIS class system lives within the overall UNIX environment and must abide by UNIX's file-level security. It may be necessary to change the UNIX protection on MANTIS system files to grant access to additional users or to protect the files from unauthorized access. MANTIS is delivered with a protection mask of rwxr-xr-x on all files. This allows the owner to access the files in any manner, but restricts groups to read and execute, and others to execute only.

The following example would change the TEST MANTIS File so that anyone on the system could access the MANTIS class:

```
# chmod 777 $MANTIS_ROOT/data/man_TEST*
```

Similarly, the MANTIS file is initially owned by the MANTIS administrator's account and group. To maintain system security, it may be necessary to give ownership of the files to the UNIX users or groups that will access this file. The following command would change the ownership of the TEST class MANTIS_FILE to a user named test and a UNIX group named test_group:

```
# chown test $MANTIS_ROOT/data/man_TEST*
# chgrp test_group $MANTIS_ROOT/data/man_TEST*
```

The UNIX command

```
# ls -l
```

will show the protection masks of files. Any problems arising with these commands should be resolved by consulting the UNIX reference manuals included with the machine.

Troubleshooting your installation

Common problems and solutions to MANTIS installation problems are listed below.

Problem The tar command stops before the final diskette(s) has been read in.

Solution This is caused by a bug in the tar command that causes tar to stop if a file ends very near the end of the input medium. This can be circumvented by placing the next distribution floppy in the disk drive, and reentering the tar command that was used to read the tapes.

Problem A UNIX error message such as “/bin/ksh: mant: not found” or “sh: mant: cannot open” is returned when the mant command is issued.

Solution This message is usually caused by either MANTIS_ROOT being defined improperly or the $MANTIS_ROOT/bin directory not being included in the user’s PATH variable. The following commands show the MANTIS_ROOT and UNIX PATH environment variables:

```
# echo $MANTIS_ROOT
/cincom/mantis_2801
# echo $PATH
/usr:/usr/bin:/cincom/mantis_2801/bin
```

The contents of the PATH variable vary from user to user and from system to system, but the variable must contain the bin subdirectory of MANTIS_ROOT.

Problem When typing mant to run MANTIS, the following message displays:

```
MANTIS Fatal Error <100>
```

Solution This indicates that MANTIS cannot access the MANTIS File. This can be caused by a missing MANTIS File or by improperly set UNIX security that prevents the user from accessing the files. The first step is to check the MANTIS_CLASS variable to see what class is being accessed. The following command shows how to determine this:

```
# echo $MANTIS_CLASS
TEST
```

Then, the class name can be used with the UNIX ls -l command to verify the existence and protections that are set on the MANTIS File. The following command could be used to check out the TEST class:

```
# ls -l $MANTIS_ROOT/data/man_TEST*
-rw-rw-rw   1  mantis  mantis      128  Sep 25 08:56 /cincom/mantis_2801/data/man_TEST.ki
-rw-rw-rw   1  mantis  mantis    19720  Sep 25 08:56 /cincom/mantis_2801/data/man_TEST.nx
-rw-rw-rw   1  mantis  mantis   210084  Sep 25 08:56 /cincom/mantis_2801/data/man_TEST.mp
-rw-rw-rw   1  mantis  mantis  2880711  Sep 25 08:56 /cincom/mantis_2801/data/man_TEST.st
    (1)          (2)      (3)
```

Pay particular attention to the privilege masks column (1), the file owner account column (2), and the file group owner column (3). These three file attributes control UNIX file security and must be set up properly before the MANTIS class can be accessed. There should also be four files shown, one containing each of the MISAM suffixes (*.ki*, *.nx*, *.mp*, and *.st*).

Problem MANTIS starts when the mant command is issued, but the screen is garbled.

Solution This is usually the result of a terminal setting that MANTIS does not recognize. Check the TERM and MANTIS_TERM environment variables to make sure they match the terminal that is being used.

Cincom attempts to include as many terminal definitions as possible in the MANTIS system. In most cases, access to new terminals is possible without altering the MANTIS file system. If this is necessary, contact Cincom Support for assistance.

Problem MANTIS hangs when the mant command is issued.

Solution This is often caused by an incompatible NFS file system. Some NFS systems cannot cope with the MISAM file systems calls. If MANTIS hangs before you get a sign-on screen, and the MANTIS File is on an NFS disk, move the MANTIS Files to local disk space. If this resolves the problem, you cannot run MANTIS on the current NFS configuration.

Problem The error message “Database needs restarted” displays when attempting to access SUPRA.

Solution This can be caused by:

- Invalid MANTIS_SQL_DBNAME environment variables.
- A version of SUPRA Server that this version of MANTIS does not support.
- Missing or invalid SUPRA DBROOT environment variable. (Refer to the SUPRA Server documentation for an explanation of the DBROOT variable.)

2

MANTIS run-time environment commands

This chapter describes the MANTIS run-time environment commands.

cp_class

The cp_class command creates a new MANTIS class from an existing class.

Each MANTIS class has a defined MANTIS Transfer File. Multiple classes can share the same Transfer File. If the Transfer File specified does not exist, it will be created.

cp_class [-h] *old_class new_class* [*xfer_file*]

-h

Description *Optional*. Prints help on standard output.

old_class

Description *Required*. Specifies the name of the MANTIS class that you want to copy.

Format 1–4 alphanumeric characters

new_class

Description *Required*. Specifies the name of the new class to be created.

Format 1–4 alphanumeric characters

xfer_file

Description *Optional*. If the new class is to have a different MANTIS Transfer File from the old class, specify it in this parameter.

Consideration If the specified Transfer File does not exist, it will be created.

Example To create an initial class, name it TEST, and assign it to use a Transfer File named TXFR, you could issue the following command:

```
# cp_class 2801 TEST TXFR
```

cp_misam

The cp_misam command copies MISAM (MANTIS Indexed Sequential Access Method) files. A MISAM file on UNIX is implemented as four separate files with suffixes (*.ki*, *.nx*, *.mp*, and *.st*). This command moves all four.

cp_misam [-h] *source target*

-h

Description *Optional*. Prints help on standard output.

source

Description *Required*. Specifies the source MISAM files.

Consideration Do not specify a file suffix (*.ki*, *.nx*, *.mp*, or *.st*).

target

Description *Required*. Specifies the target MISAM files.

Considerations

- Do not specify a file suffix.
- You can specify only a directory.

Example To copy a MANTIS MISAM file named TEST to a MISAM file named TST2, you would issue the following command:

```
# cp_misam TEST TST2
```

mant

The mant command runs MANTIS using the current MANTIS class.

mant [=h] [*mantis_arg* ...]

=h

Description *Optional*. Prints help on using the mant command.

mantis_arg

Description *Optional*. Specifies any arguments to MANTIS.

Consideration All command line arguments will be passed on to MANTIS, except for =h.

mant_mgr

The mant_mgr command provides a menu system for the person who manages the MANTIS system. It provides an easy-to-use front end for the other commands described in this chapter.

mant_mgr [-h]

-h

Description *Optional*. Prints help on standard output.

General consideration

- You should start out using the mant_mgr program to perform maintenance on the MANTIS system. Then, as you become more familiar with the system, switch to using the commands directly. Each item on the menu indicates which command (cp_class, rm_class, etc.) it calls.

mcpy

The mcpy command runs the MANTIS mcpy (MANTIS Copy) utility to merge two MANTIS Files or to upgrade from MANTIS release 2.7 to MANTIS release 2.8, or to downgrade from MANTIS release 2.8 to MANTIS release 2.7.

mcpy [=h] *in_dir in_class out_dir out_class*

=h

Description *Optional*. Prints help on using the mcpy command.

in_dir

Description *Required*. Specifies the directory in which the input MANTIS File is located.

in_class

Description *Required*. Specifies the name of the input MANTIS File.

out_dir

Description *Required*. Specifies the directory in which the output MANTIS File is to be placed.

out_class

Description *Required*. Specifies the name of the output MANTIS File.

General consideration

- All command line arguments will be passed on to mcpy, except for =h.

mfm

The mfm command runs the MISAM file maintenance utility. It takes as input the name of a MISAM file and repairs and rebuilds it. Note that a MISAM file on UNIX consists of four separate files—a store file (.st), an index file (.nx), a mapper file (.mp), and a key information file (.ki).

The MISAM File Maintenance utility first clears the MISAM file lock list (unlocking the file if it is locked). The definition (.def) file describing the MISAM file attributes is created next. The information in the definition file is used in the subsequent repair/rebuild operation. The store file data is checked, repaired, and compressed; the mapper file delete thread list is reconstructed; the index file is rebuilt; and record counts are checked to be consistent.

mfm [=h] *misam_file*

=h

Description *Optional*. Prints help on using the mfm command.

misam_file

Description *Required*. Specifies the file to be repaired.

Consideration Do not specify suffixes on the MISAM file name.

mhlp

The mhlp command runs the MANTIS mhlp (MANTIS Help) utility using the current MANTIS class.

mhlp [=h] [*mhlp_arg* ...]

=h

Description *Optional*. Prints help on using the mhlp command.

mhlp_arg

Description *Optional*. Specifies any arguments to mhlp.

Consideration All command line arguments will be passed on to mhlp, except for =h.

minstall

The minstall command installs a MANTIS run-time environment from a restored release tape. Before running minstall, set your working directory to the $MANTIS_ROOT directory. Before the installation starts, you will be asked to confirm that the current directory is $MANTIS_ROOT.

sh ./minstall [-h]

-h

Description *Optional*. Prints help on standard output.

General consideration

- For this command to work properly, the current directory must be included in your PATH environment variable. So that older versions of installation shell scripts will not be picked up by mistake, your PATH should not reference a previous release of MANTIS for UNIX.

mk_setup

This script reads the current UNIX user environment and generates a start-up script for a class, based on that environment. It reads any of the MANTIS environment variables that exist and creates a script to recreate those variables for a new class. If any variables do not exist, commented example lines are included in the generated script.

The script is placed in a file called *xxxx*_SETUP, where *xxxx* is the class name passed to the script.

mk_setup [-h] *class*

-h

Description *Optional*. Prints help on standard output.

class

Description *Required*. Specifies the MANTIS class for which the script is to be generated.

General consideration

- The generated script is placed in the current directory.

mop

The mop command runs the MANTIS mop utility using the current MANTIS class.

MANTIS mop compiles a MANTIS options source file that will customize the MANTIS environment for your needs. Refer to *AD/Advantage MANTIS Administration Open VMS/Unix*, P39-1320, for details on available options.

mop [=h] [*mop_arg* ...]

=h

Description *Optional*. Prints help on using the mop command.

mop_arg

Description *Optional*. Specifies any arguments to mop.

Consideration All command line arguments will be passed on to mop, except for =h.

mpr

The mpr command runs the MANTIS mpr utility using the current MANTIS class.

The MANTIS mpr utility is used to extract descriptions of MANTIS entities into printable text files. For details on how to use mpr, refer to *AD/Advantage MANTIS Administration Open VMS/Unix*, P39-1320.

mpr [=h] [*mpr_arg* ...]

=h

Description *Optional*. Prints help on using the mpr command.

mpr_arg

Description *Optional*. Specifies any arguments to mpr.

Consideration All command line arguments will be passed on to mpr, except for =h.

rm_class

The rm_class command deletes the specified MANTIS class. The MANTIS File is deleted; however, the Transfer File is not deleted because it may be used by other classes. Use rm_misam to remove the Transfer File.

rm_class [-h] *class_name*

-h

Description *Optional*. Prints help on standard output.

class_name

Description *Required*. Specifies the name of the MANTIS class to be deleted.

Consideration The Transfer File is not removed.

Example To remove a MANTIS class named TEST, you would issue the following command:

```
# rm_class TEST
```

rm_misam

The rm_misam command deletes MISAM files. A MISAM file on UNIX is implemented as four separate files. This command deletes all four.

rm_misam [-h] *file*

-h

Description *Optional*. Prints help on standard output.

file

Description *Required*. Specifies the source MISAM file.

Consideration Do not specify a suffix.

Example To remove a MISAM file named TEST, you would issue the following command:

```
# rm_misam TEST
```

A

Sample MANTIS installation (minstall) output

This appendix shows the output from an actual installation under SCO UNIX.

Please note the errors near the end of the installation. These errors occur on some of the platforms, due to limitations in the TERMINFO facilities. These tic warning messages indicate that the operating system you are running is incapable of supporting some terminal features. These messages can be safely ignored in most cases. However, it would be a good idea to save a listing of the errors, to assist Cincom support staff, in the event that terminal problems do arise.

```
# sh ./minstall
------------------------------------------------------------
AD/Advantage - MANTIS RUNTIME ENVIRONMENT (MRE) INSTALLATION PROCEDURE
------------------------------------------------------------
This is the installation procedure for installing the
AD/Advantage - MANTIS Runtime Environment (MRE) on UNIX.

This procedure will create the MRE.

All MRE files are stored in a directory tree of fixed structure.

  Uncompressing AD/Advantage files in $MANTIS_ROOT/bin ...

  Uncompressing AD/Advantage files in $MANTIS_ROOT/lib ...

  Uncompressing AD/Advantage files in $MANTIS_ROOT/exe ...

  Uncompressing AD/Advantage files in $MANTIS_ROOT/data ...

You can install an AD/Advantage - MANTIS Class (default) or a MANTIS only class.  If
you are not licensed for AD/Advantage you must install MANTIS only.

Do you wish to install MANTIS only (y/n) [n] ?

  Creating clean MANTIS class, 2801 ...

  Creating MANTIS termtable ...

tic:  Warning:  near line 5:  terminal 'ba80-08|ba80-vt220|ba80-Variante 03im VT220-
Modus + 132 Spalten+ ct08-Tastatur', Unknown Capability - 'iso'

tic:  Warning:  near line 5:  terminal 'ba80-08|ba80-vt220|ba-Variante 03 im VT220-
Modus + 132 Spalten+ ct08-Tastatur', Unknown Capability - 'skz'

.
.     Many Warning lines omitted for printing purposes
.

tic:  Warning:  near line 26: terminal 'ba80-08|ba80-vt220|ba80-Variante 03 im VT220-
Modus + 132 Spalten+ ct08-Tastatur', Unknown Capability - 'skz'

tic:  Warning:  near line 95:  terminal '2392-rev|2392a-rev|2392A-rev|hp2392-
rev|hp2392a-rev|hp2392A-rev|hp 2392/2622 series', Unknown Capability - 'menu'

------------------------------------------------------------
AD/Advantage - MANTIS MRE INSTALLATION PROCEDURE COMPLETE
------------------------------------------------------------
```

B

MANTIS for UNIX authorization

This chapter provides a brief introduction to the MANTIS product authorization system, which controls the use of MANTIS and its components through an encrypted security code. Security codes can be activated by the following methods:

- **Built-in codes.** MANTIS has one set of codes built into the product when it is compiled. These codes activate the product for a very short period and are typically used to control trial tapes.
- **MANTIS_PATCH environment variable.** The second method for activating codes is to assign the encrypted key to a UNIX environment variable named MANTIS_PATCH.

  The following is an example line of a MANTIS_PATCH environment variable:

  ```
  MANTIS_PATCH='01234567 01234567 012345678 01234567
  01234567 01234567 01234567 01234567' ;export MANTIS_PATCH
  ```

  The patch code contains the following information:

  - Dates when the MANTIS product is authorized to operate.
  - MANTIS components that are authorized to operate.

MANTIS begins to issue warnings 30 days before code expiration. Contact Cincom Support if these messages appear when signing on to MANTIS.

C

Platform-specific installation notes

This chapter contains installation considerations for SCO UNIX and SUN-OS environments.

SCO UNIX

To run MANTIS on the SCO system console, set TERM to “ansi” and MANTIS_TERM to “sco_console” for best results.

```
TERM=ansi # export
MANTIS_TERM=sco_console
```

If MANTIS will be used in an SCO environment with the SCO INTERNATIONALIZATION support set to use the European style decimal points, the following environment variable must be created in the user’s current shell:

```
#export LC_NUMERIC=english_us.8859
```

This forces SCO to override the internationalization setting for the decimal point, allowing the built-in MANTIS decimal conversion support to work.

SUN-OS

To run MANTIS on the SUN workstation, set TERM or MANTIS_TERM to “sun-cmd” or to “sun” if running in a shelltool window under OpenWindows.

```
TERM=ansi # export
MANTIS_TERM=sun_console
```

To run in SunView, MANTIS requires a .ttyswrc file to define all function key escape sequences that should be passed to MANTIS. A file, .ttyswrc, is supplied in $MANTIS_ROOT/data for this purpose. It defines the escape sequences for the SUN workstation keys that are used by MANTIS. Copy this file to the home directory of each MANTIS user. Note that other software you have installed on your system may also require .ttyswrc files in order to operate correctly. You must exit SunView to change .ttyswrc files to use these other applications.

When using SunView, insert the following entry in your home directory .defaults file if you want bold text to be displayed as bold rather than reverse video. Please note that case is important.

```
/Tty/Bold_style "Offset_X"
```

When using OpenWindows, insert the following entry in your home directory .Xdefaults file if you want bold text to display as bold rather than reverse video. Please note that case is important.

```
term.boldStyle:  Offset_X
```

Currently, the MANTIS run-time environment requires UNIX System V utilities. On SUN-OS systems, this means that to install or use MANTIS, your path must include /usr/5bin before /usr/bin—for example,

```
PATH=/usr/5bin:$PATH; export PATH
```

D

New features and changes in service levels

Service level 2601

This section describes the new features and changes in service level 2601.

New MANTIS options

The following MANTIS options were added in service level 2601:

Boolean options

- **AUTOFORMFEED**. Enables output of form feed characters between pages of text written to the PRINTER in scroll mode.
- **AUTOREFRESH**. Enables automatic screen refreshing after a CALL of PERFORM statement.
- **CISAMTM**. Enables transaction recovery support for CISAM access methods. This is a product licensing option and is authorized in the product security patch provided by Cincom.
- **EXTCURSORON**. Enables cursor visibility during execution of CALL and PERFORM statements. Use the NOEXTCURSORON option to make the screen cursor invisible while your external interface or command procedure is executing.
- **LEADFORMFEED**. Enables output of a leading form feed character to PRINTER files. Use the NOLEADFORMFEED option to prevent MANTIS from outputting a leading form feed character.

- **ORASQL**. Enables support for Oracle SQL. This is a product licensing option and is authorized in the product security patch provided by Cincom.

- **READONLY**. Read-only MANTIS file access. Use this option to make sure you do not update the MANTIS file. You have to specify READONLY in the command line when invoking MANTIS or in an options command file executed during MANTIS initialization.

- **REALM**. Enables read-only access to ULTRA data sets when the ULTRA statements specify a read-only password. Use the NOREALM option to force MANTIS to open PDM files for read/write access regardless of the passwords specified in ULTRA statements.

Numeric option

- **EZONE**. Specifies the screen field width used in determining whether to display numeric data in scientific notation or fixed-point decimal notation. EZONE only applies to unmasked numeric screen fields. Set EZONE=13 for compatibility with earlier versions of MANTIS, and for compatibility with MANTIS for the IBM mainframe.

String option

- **SCREENTAG**. Specifies a character string to be displayed on every MANTIS CONVERSE and on every full screen refresh performed automatically by MANTIS.

Deleted MANTIS option

The USERMIXED option was deleted in service level 2601.

New data types

MANTIS 2.6 supports INTEGER and DECIMAL data types internally and externally if the external database permits it. You can create INTEGER and DECIMAL variables using MANTIS design facilities, and/or declare them in a MANTIS program using the following statements. Refer to *AD/Advantage MANTIS Language OpenVMS/UNIX*, P39-1310, for a detailed specification of these statements and more detailed numeric considerations.

DECIMAL [(*precision*)]*decimal-name*[(*dimension*,...)],...

- DECIMAL numbers are represented internally in ASCII format and support up to 31 significant digits (precision). There is no loss of accuracy representing definite decimal numbers as there sometimes is with a floating point (BIG and SMALL).
- DECIMAL numbers have an exponent like BIG and SMALL numbers. The range of the exponent is -128 to +127.
- DECIMAL maps naturally to external PACKED and ZONED data types, especially when more than 14 significant digits are involved.
- Scientific notation, previously limited to E-notation, had been extended to allow D-notation. For example, the numeric constant 1.1D-1 represents the value 0.11, but this value will be stored internally (and accurately) in DECIMAL format.
- DECIMAL arithmetic is performed by software and is much slower than binary and binary floating-point arithmetic.

INTEGER *integer-name*[(*dimension*,...)],...

- INTEGER values are signed whole numbers in the range -2147483648 to 2147483647. They are stored internally in a binary longword (4 bytes).
- INTEGER maps naturally to external BINARY data types.
- This data type is more efficient than other numeric data types for simple counting and indexing operations.

Screen design enhancements

New field attributes in screen designs give MANTIS 2.6 improved field editing capabilities during a CONVERSE. Refer to *AD/Advantage MANTIS Facilities OpenVMS/UNIX*, P39-1300, for more information.

Field entry routines

In MANTIS 2.5, a field validation routine was a MANTIS subroutine invoked as a result of modifying a screen field during a CONVERSE. The routine could be invoked dynamically by setting the Field Sensitive Validation attribute. In MANTIS 2.6, this type of validation routine is called a field exit routine. In addition to defining field exit routines, MANTIS 2.6 allows you to define the field entry routines to invoke MANTIS code whenever a field gains the input focus (when the cursor enters the field).

The name of the field entry routine can be entered in Screen Design on the Update Field Specifications screen. A field entry routine is not considered to be a validation routine because data validation is typically a function performed after changes have been made to a field. For example, field entry routines can be used to:

- Highlight the field with input focus. The highlighting can be reversed by a matching field exit routine (provided Field Sensitive Validation is enabled).
- Highlight fields related to the field gaining input focus. Sometimes a change to one field requires data entry in one or more related or dependent fields. You can highlight those related fields as a visual aid to data entry requirements.
- Validate the correct sequence of data entry by verifying that prerequisite fields have been entered correctly.
- Supply default values for a field. Normal methods of specifying default values are sometimes inadequate. The default value of a field may depend on what has already been entered in other screen fields. The visual effects of this may also be desirable.

Field entry and exit routines take the same parameters. MANTIS 2.6 provides one more parameter to enable the sale MANTIS subroutine to be used for both purposes:

ENTRY *routine-name* [(p1[,p2[,p3[,p4[,p5[,p6[,p7[,p8]]]]]]])]

The p8 SMALL parameter is set to TRUE for a field entry routine and FALSE for a field exit routine.

Forced Field Sensitive Validation

To understand Forced Field Sensitive Validation, it is necessary to review Field Sensitive Validation first.

The Field Sensitive Validation (FSV) attribute causes MANTIS to validate changes to a field when the field loses the input focus. MANTIS executes whatever validation attributes are defined, and if the field is in error it places the cursor back in the field and displays an error message. In previous releases of MANTIS, this only happens if the contents of the field are modified by data entry. That is, you can tab into and out of an FSV field without MANTIS executing any validation code.

MANTIS 2.6 provides the FFV attribute (Forced FSV), which ensures that all field validation logic executes whenever the field loses input focus, whether or not the user modified it.

For example, you can use FFV to:

- Reset field attributes that may have been set by a field entry routine.
- Enforce your own tab-stop order for data fields. Use a common field exit routine to make tab stops proceed down the screen instead of across the screen. This is useful for editing columns of data. You can also choose to skip one or more input fields if the last field was not altered.

Enhanced UPDATE statement

This section provides an overview of changes to the UPDATE statement. Refer to *AD/Advantage MANTIS Language OpenVMS/UNIX*, P39-1310, for more information.

UPDATE *screen-name*

Use this form of the UPDATE statement to refresh the logical display immediately with the latest data values belonging to *screen-name*.

UPDATE is quicker than CONVERSE with TIME=0, and may be used in a field entry or field exit routine.

Facilities enhancements

The following change has been made to MANTIS Facilities. Refer to *AD/Advantage MANTIS Facilities OpenVMS/UNIX*, P39-1300, for more information.

MANTIS Facilities have been modified to allow DECIMAL and INTEGER field definition. In External File View Design, the Update Record Layout screen is changed to allow you to enter the internal MANTIS data type for each field. Previously, this field was protected and MANTIS chose the internal data type based on the specified external data type (binary, packed, etc.). With the extra choice of internal data type, MANTIS no longer forces a particular type. If you only enter one data type (internal or external), MANTIS supplies a suitable default for the other type.

Batch MANTIS scripts used to create record layouts must be modified since the internal MANTIS TYPE field is now unprotected.

Extended internal interface support

An internal interface program is one that MANTIS dynamically link-loads when the Call statement is executed. External interface programs are executed by MANTIS in a separate system subprocess.

Internal interface support is only available on selected platforms. This version supports the HP/UX, AIX, and OSF/1 UNIX platforms.

Universal Export Facility

The Universal Export Facility (UEF) Options screen contains one new option governing the output of passwords during export of programs, file views (FILE, ACCESS, and ULTRA), interfaces, and prompters. These entities are either password-protected or contain access passwords (in the case of views). For security reasons, you may want to omit passwords from external export files.

Omitting passwords on export means they are also lost on subsequent imports, potentially changing application functionality.

P-code performance improvements

Improvements include stabilization of the P-code interpreter, fixes, enhancements, and binding of all CONTROL programs in the distributed MANTIS file.

Dynamic load DATABAS

This feature allows dynamic loading of the PDM interfaces on UNIX systems, where possible. This eliminates the need to install new releases of MANTIS when new release/service levels of PDM are installed.

Terminal definition facility

For UNIX users, the new MTI (MANTIS Terminfo) utility provides a user-friendly facility for defining custom terminal keyboards to MANTIS for UNIX.

This utility also provides a new MANTIS parameter on the GET statement that will use a PDM secondary key, allowing MANTIS to issue a READX command. This has been implemented to be consistent with MANTIS for WANG/VS and MANTIS for NCR/VAX, so that MANTIS applications developed on these platforms are now portable to UNIX platforms where PDM indexing is available.

Miscellaneous maintenance and bug fixes

Existing bug fixes from MANTIS 2.4 and 2.5 releases have been implemented in source. Additionally, several problems that could not be corrected through a patch have been corrected in MANTIS 2.6.

Service level 2701

This section describes the new features and changes in service level 2701.

In IBM compatibility mode, MANTIS 2.7 is compatible with MANTIS for the IBM mainframe, release 5.2 and 5.4. You may use the UEF Facility to transfer MANTIS user entities and data between MANTIS 2.7 and MANTIS for the IBM mainframe.

Year 2000 support

This release includes support for 4-digit year dates when using the DATE function to return a text string containing the current date. The format of the date is determined by a previous date mask specification using the DATE statement. The default is *YY*/*MM*/*DD*.

All MANTIS Facilities functions will use 4-digit dates for all internal date fields. Any dates displayed by MANTIS in any of the Facility screens are displayed in *YYYY*/*MM*/*DD* format.

Year 2000 test window

This release includes a Year 2000 test window. MANTIS product authorization allows a window for “setting the clock forward.” If you have a valid expiration date, MANTIS will permit the system clock to be advanced to any time between December 1, 1999 through April 15, 2000. This will allow for testing year-end, quarter-end, and month-end processing before, during, and after year 2000 rollover.

New MANTIS options

The following MANTIS options were added in service level 2701.

Boolean options

- **IGNORETRLBLKS**. Enables MANTIS to ignore trailing blanks when the STRCOMPARE option is also set to ALPHANUMERIC for a given program.
- **ORAUNPAD**. Specified if MANTIS text variables that are used as Oracle input host variables will have trailing blanks removed.

Numeric option

- **DECUSELEN**. Specifies the number of significant digits within a constant that a BIG data type will be encoded as a DECIMAL constant.

Fixed List option

- **NULLPOSITION**. This option has been modified to allow a setting of 2. In MANTIS/VAX 1.0, POINT (string-NULL) returns 1. Other releases of MANTIS return 0. When NULLPOSTITION is set to 2, MANTIS will return 1 to POINT function, the same as MANTIS/VAX 1.0.

Extended internal interface support for UNIX

An internal interface program is one that MANTIS dynamically link-loads when the CALL statement is executed. External interface programs, on the other hand, are executed by MANTIS in a separate system subprocess.

Internal interface support is only available on selected platforms. For MANTIS 2.7 this support has been extended to the Sun/Solaris UNIX platform, to go along with AIX, HP-UX and OSF/1 UNIX platforms.

Unloading of dynamically linked internal interface programs for HP UNIX

For internal interface programs running on HP/UX UNIX only, MANTIS has been modified to allow the dynamic unloading of internal interface programs (.sl) which are dynamically loaded by MANTIS. This is indicated to MANTIS by setting the internal interface VIEW field Internal Routine Resident to N. This informs MANTIS to dynamically unload the routine when the routine returns to MANTIS. The default setting is Y for the routine to remain resident.

Oracle array-processing support

Oracle SQL support for MANTIS has been enhanced to support array processing. This allows one SQL statement to process multiple SQL rows. MANTIS numeric and text arrays can be used as Oracle SQL host variables. The FOR SQL keyword is supported to allow specification of the number of rows. See the Oracle *SQL Language Reference Manual* for more details.

Increase resource table

Previously the MANTIS Resource Table had a fixed size of 250. Now this value has been placed in the updated MASTER Facility, Update Shared Entity List, option 8, Set Global Resource Quota. For Release 2.7, the Set Global Resource Quota screen now contains the following fields:

- **MAXUSERS.** This value represents the total number of concurrent MANTIS processes running against a particular MANTIS CLASS. The default is 1000. The minimum is 50; the maximum is 9999.

- **MAXCONCURRENTENQ.** This value represents the maximum number of concurrent ENQUEUE statements upon a single resource name of all MANTIS users running against a particular MANTIS CLASS. For example, if MAXUSERS was set to 100 and each user was running the same program which contained an ENQUEUE “PAYROLL,” KEY statement, then theoretically the MAXCONCURRENTENQ value might be set as high as 100. But a more practical value might be 20, meaning that at MAX, not more than 20 MANTIS users would be waiting for that ENQUEUE at one time. The default is 50. The minimum is 10 and the maximum is 5000.

These values are then used to calculate the size of the Resource Table in shared memory.

For these values to take effect, all users for this MANTIS CLASS must terminate their MANTIS process. Once all MANTIS processes for this CLASS have been terminated, the users can restart the MANTIS process to take effect.

If a MANTIS process must be terminated by the UNIX kill command, the preferred signal would be 16; for example, kill -16 ####. This would allow MANTIS the opportunity to clean up any resources allocated (shared memory and semaphores) that the MANTIS process created. If for some reason the MANTIS process does not terminate, kill -9 will terminate the process. In this case, MANTIS will not have the opportunity to clean up the resources created by that process. To clean up those resources, use the UNIX ipcs and ipcrm commands. An easier approach would be for the UNIX user whose process was terminated to rerun MANTIS. This would allow the new MANTIS process to reclaim any lost or delinquent resource and remove them upon MANTIS termination of that process.

The default settings for MAXUSERS and MAXCONCURRENTENQ are adequate for most installations.

Removal of Sticky Bit on MANTIS executable

With the redesign of the MANTIS Global Resource in this release, the need for setting of the Sticky Bit (set user ID on execution) is no longer needed for MANTIS release 2.7.

Dynamic load DATABAS

MANTIS release 2.7 allows dynamic loading of the SUPRA Server PDM interface on all UNIX systems where SUPRA Server PDM is supported. This eliminates the need for customers to install new releases of MANTIS when new release/service levels of SUPRA Server PDM are installed.

Extended CISAM and CISAM-TM support for UNIX

CISAM and CISAM-TM for external files support is only available on selected platforms. For MANTIS 2.7 this support has been extended to the OSF/1, Sun/Solaris, DG-UX, and NCR3000 UNIX platforms, to go along with AIX, HP-UX, SCO, and Sun/OS platforms.

The version of CISAM/CISAM-TM for each UNIX platform has also been upgraded to the latest available release from Informix.

Miscellaneous maintenance and bug fixes

Existing bug fixes from MANTIS 2.6 have been implemented in source. Additionally, several problems that could not be corrected through a patch have been corrected in MANTIS 2.7

Service level 2801

This section describes the new features and changes in service level 2801.

In IBM compatibility mode, MANTIS 2.8 is compatible with MANTIS for IBM mainframe, Release 5.4. You may use the UEF (Universal Export Facility) to transfer MANTIS user entities and data between MANTIS 2.8 and MANTIS for the IBM mainframe.

MANTIS Search Facility (MSF)

The MANTIS Search Facility (MSF) is a programmer's tool to search through the text of MANTIS entities and find matches for supplied criteria.

MSF searches Program Lines for supplied criteria, including optionally searching comments within a Program. Other options include case sensitivity on the searching of the supplied criteria. Other entities are searched for matching field names. These entities include Screens, Internal Files, External File Views, Prompters, Interfaces and Ultra/PDM File Views.

Additionally, the entity description field can be searched. The results of the MSF search can be directed to either the user terminal and/or printer, at a report level or page level.

Additional Universal Export Facility support (UEF)

The MANTIS Search Facility (MSF) "Search Save Files" has been added to the entity types that can be exported and imported through the Universal Export Facility.

List of current MANTIS users

The List of Current MANTIS Users is a MASTER User Facility that is used to display all MANTIS Users current on the MANTIS System.

The List of Current MANTIS Users will list such items as the number of User and Developer Seat Licenses available, along with the MANTIS User Name, MANTIS Class, and the Logged in Time and Date. UNIX information is also displayed per MANTIS User, such as the MANTIS Executable UNIX Process_Id, UNIX User Name along with the User_Id and the Group_Id for that user. The UNIX TTY for the terminal being used by MANTIS is also displayed.

Display of MANTIS security patch information

The Display of MANTIS Security Patch Information is a MASTER User Facility that is used to display the current MANTIS_PATCH environment settings.

The display of MANTIS Security Patch Information will list such items as the Customer_Id/Account Number, the Security Patch Expiry Date, the number of User and Developer Seat Licenses, the authorized Sign on Access to Protected Users, the authorized Read Access to Protected Libraries, along with the list of Authorized MANTIS Product Options.

New MANTIS options

- BOXGRAPHICS
- NOBOXGRAPHICS

Description: Specifies whether box graphics characters contained within MANTIS Screens should be displayed or converted to simulated box characters contained within the BOXCHAR option during a MANTIS CONVERSE Statement.

Performance improvements

Redesign of the “mant” script

The “mant” script was redesigned not to produce any Temp Files in the \tmp directory. This script used to produce 6 Temp Files in the \tmp directory per MANTIS process, will now produce none. This redesign will free up system resources of Temp Files being created and deleted by the system. This will allow MANTIS to initialize faster at start up time and de-initialize faster at shutdown time.

Example: For a 60 User Site, the system would create and delete a total of 360 Temp Files in a normal processing day. Now the system has no Temp Files to maintain.

Redesign of the ‘mant_mgr” script

The “mant_mgr” script was redesigned to reduce the number of Temp Files in the \tmp directory from 11 down to 2, per MANTIS process. This redesign will free up system resources of Temp Files being created and deleted by the system.

Example: For a 60 User Site, the system would create and delete a total of 660 Temp Files in a normal processing day. The total is now reduced to 120 Temp Files.

Redesign of the Global Shared Memory Allocation

This release of MANTIS will now allocate the amount of Global Shared Memory based on the MANTIS Security Patch for the number of MANTIS Seat Licenses. For previous releases, MANTIS allocated 1.2 Megabytes of Shared Memory per MANTIS_CLASS. In this release of MANTIS, for a 60 User Site, MANTIS reduces the memory requirement to 170 Kilobytes per MANTIS_CLASS. This results in a saving of over one Megabyte per MANTIS_CLASS.

Miscellaneous maintenance and bug fixes

Existing bug fixes from MANTIS 2.7 have been implemented in this source. Additionally, several problems that could not be corrected through a patch have been corrected in MANTIS 2.8.

Index

N

O

P

R

S

T

U

V

Y